

A Comparative Study on Contour-based Corner Detectors

Mohammad Awrangjeb¹, Guojun Lu² and Clive S. Fraser¹

¹Department of Geomatics, The University of Melbourne
CRC for Spatial Information, 723 Swanston St, Carlton Vic 3053 Australia

{mawr, c.fraser}@unimelb.edu.au

²Gippsland School of Information Technology, Monash University

Guojun.Lu@infotech.monash.edu.au

Abstract

Contour-based corner detectors directly or indirectly estimate a significance measure (e.g. curvature) on the points of a planar curve and select the curvature extrema points as corners. While an extensive number of contour-based corner detectors have been proposed over the last four decades, there is no comparative study of recently proposed promising detectors. This paper is an attempt to fill this gap. We present the general frame-work of the contour-based corner detection technique and discuss two major issues – curve smoothing and curvature estimation, which have major impacts on the corner detection performance. A number of promising detectors are compared using an automatic evaluation system on a common large dataset. It is observed that while the detectors using indirect curvature estimation techniques are more robust, the detectors using direct curvature estimation techniques are faster.

1. Introduction

The terms ‘dominant point’, ‘critical point’ and ‘corner’ are used as equivalents in the literature to indicate the detected corners on a planar curve (open or close contour) [29]. Although the notion of a ‘corner’ seems to be intuitively clear, no generally accepted mathematical definition exists for digital curves [11]. In fact, different approaches give different but conceptually related computational definitions to a visual phenomenon. For example, Guru et al. [12] defined a corner as the intersection of two adjacent relatively straight curve-segments, i.e. That is a corner point is found at a location where the direction of the curve changes significantly and abruptly.

Although the terms ‘interest-point’ and ‘corner’ are sometimes used equivalently in the literature, corners are a special subclass of interest-points. Interest-points include not only corners, but also T-junctions, blobs as well as lo-

cations of significant texture variations. In general, corners offer many advantages over interest-points [4]. Firstly, corners are visually distinguishable and more robust than their interest-point counterparts. Secondly, in an image, the number of corners is much lower than that of interest-points. This can significantly speed up feature matching algorithms if corners are used. Thirdly, corners can be ranked based on their strength like the curvature value, or the number of corners can be controlled by changing the detection thresholds. In contrast, it is very difficult to rank interest-points since their descriptors are of higher dimension. Finally, corner detection requires less time than interest-point detection.

Contour-based corner detectors, in general, either use a polygonal approximation to search for intersection points or look for curvature maxima points along the extracted contours from the gray-scale image. Algorithms using polygonal approximation obtain a piecewise linear polygonal approximation of the curve subject to some constraints on the goodness of fit. The actual or extrapolated intersections of the adjacent line segments in the approximated polygon are then detected as dominant points [19]. Algorithms using curvature estimation use a significant measure, e.g. curvature [22, 23], on the planar curve. They then look for curvature extrema points as corners. This paper focuses on these detectors.

Recently, a number of promising contour-based detectors have been proposed. However, a comparative study to judge them on a common platform has not as yet been undertaken. This paper is an attempt to fill this gap. Note that the earlier comparative work by Mokhtarian and Mohanna [16] is almost five years old. Moreover, they used a small database and their evaluation system was manual and consisted of flawed evaluation metrics [5]. In this paper, we present a comparative study of seven promising detectors using an automatic evaluation system [6] on a large database.

The rest of the paper is organized as follows. In Sections 2 and 3, we present the general frame work of contour-based

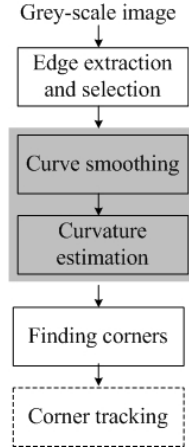


Figure 1. A flow chart for general corner detection technique.

corner detection techniques and discuss two major issues, namely curve smoothing and curvature estimation, which have major impacts on the corner detection performance. In Section 4, we first summarize the results from the literature and then a number of promising detectors are compared using an automatic evaluation system on a common large dataset. Finally, the paper is concluded in Section 5.

2. General Framework of Contour-based Corner Detection

As shown in Fig. 1, common contour-based corner detection techniques consist of five steps: edge extraction and selection, curve smoothing, curvature estimation, finding corners and coarse-to-fine corner tracking. The ‘corner tracking’ step in the dotted rectangle is an optional step and therefore is not used by many detectors. The shaded rectangles for the ‘curve smoothing’ and ‘curvature estimation’ steps indicate that these two steps are sometimes integrated, for example, for detectors estimating indirect curvature with indirect smoothing technique (see Section 3.3).

- *Edge extraction and selection:* Given a grey-scale image, a corner detector first extracts edges using a selected edge detector. Most of the corner detectors in the literature do not detail this step since edge extraction is a separate algorithm, commonly used by all corner detectors. However, Awrangjeb and Lu [5] showed that extraction of strong edges and elimination of short edges, which may be large in number, can increase the performance of a detector.
- *Curve smoothing:* Slope and curvature evaluation of a discrete curve is not simple, since the position (coordinates) of a curve-point is quantized. Moreover, there may be noise and local variation on the curve. Therefore, some smoothing should be carried out before or

during the curvature calculation in order to make the curvature extrema points more distinguishable from other curve-points.

There are two kinds of smoothing: direct and indirect. A direct smoothing, e.g. the Gaussian smoothing [17], removes noise and changes the curve-point locations, while an indirect smoothing, e.g. the *region of support* (RoS) [22] or the chord-length [19], does not change the curve-point locations at all. As the size of RoS or chord is increased, the curvature estimation technique overlooks the fine details of the curve without substantially changing the curve-point locations. Both kinds of smoothing have a similar effect in the sense that using a low value of the smoothing parameter may not reduce the effect of noise completely and may detect many spurious corners, while using a large value of the smoothing parameter may overlook important details of the curve.

Since detectors using indirect smoothing do not physically remove noise from the curve, they may detect spurious corners and thus cannot localize the corners well. Successful corner detection requires some direct curve smoothing to be performed beforehand [2]. The importance of direct curve smoothing (Gaussian smoothing with scale σ) has also been noted in [22, 23], though smoothing was not applied. Awrangjeb and Lu [6] applied a small Gaussian smoothing-scale (direct smoothing) before the estimation of curvature using chords (indirect smoothing). In addition to using a fixed Gaussian smoothing, He & Yung [13] used a dynamic (adaptive) RoS to calculate the variable curvature-threshold.

- *Curvature estimation:* A corner detector then estimates a significance measure, i.e. curvature, on each point of the smoothed curve. The estimated curvature is sometimes referred to as ‘corner strength’ or ‘cornerity’ [4]. There are two types of curvature estimation techniques in the literature: direct and indirect. In direct curvature estimation techniques, an algebraic or geometric estimation such as cosine, local curvature or tangential deflection [17, 23] on each curve-point is used as the significance measure. Such algebraic or geometric estimations are measured on each point with its neighboring points. For example, Mokhtarian and Suomela [17] estimated the Euclidean curvature (local curvature) at each point considering two neighboring points on each side.

In contrast, an indirect measure is used as the significance measure in indirect curvature estimation techniques. For example, Masood and Sarfraz [14] counted the number of curve-points lying inside three rectangles moving along the curve. Ramer [19] used the

perpendicular distances from the chord connecting the two end-points of the curve (segment) to curve-points. Some other indirect measurements of the significance measure are area [30] and distance [28] functions. Since the direct curvature estimation techniques consider a less number of neighboring points on each side of a point, they are more sensitive to noise and local variation of the curve than their indirect counterparts [6] (see detail in Section 3.2).

- *Finding corners*: A corner detector gathers the curvature maxima points on all of the smoothed curves in the candidate corner set. A maximum point can either be a strong, weak (also known as ‘round’ in the literature [17]), or false corner. The later two should not be regarded as corners.

Thresholds are applied to remove weak and false corners [6]. He & Yung [13] used a curvature-threshold to remove weak corners and an angle-threshold to remove false corners.

- *Corner tracking*: If corners are detected using high Gaussian smoothing-scales (σ), their localization is not good. A coarse-to-fine corner tracking is carried out for the detected corners to improve their localization. For example, Mokhtarian and Suomela [17] detected corners at $\sigma = 4$ and tracked them through $\sigma = 3$ and 2 to $\sigma = 1$. At each scale (say, $\sigma = 2$) the curvature maximum point around a corner (which was detected at the immediate higher scale, i.e. $\sigma = 3$) is selected as the next tracked position for that corner. The tracked position at $\sigma = 1$ is the final location for that corner.

Note that the corner tracking step does not use any threshold but changes the corner positions only, not the number of corners. This corner tracking step is optional for a number of detectors [13, 6], which use low σ values for smoothing.

3. Major Issues of Corner Detection

Curve smoothing and curvature estimation are the most critical steps in corner detection. In order to detect all true corners at their correct locations and not detect any false or weak corners on a curve, it is very important that the curve should be smoothed with an appropriate smoothing-scale. However, the selection of smoothing-scale for a given curve is not an easy task.

On the other hand, curve-point locations are quantized and there may be noise and local variations on the curve. In addition, geometric transformations or signal processing may affect the edge detection performance. As a result, it is difficult to consistently estimate curvature at a curve-point under different conditions.

Detectors estimating indirect curvature with indirect smoothing techniques usually integrate (combine) the curve smoothing and curvature estimation steps.

In this section, we discuss more on the selection of smoothing-scale and curvature estimation techniques.

3.1. Smoothing-scale Selection

A smoothing parameter, which indicates the amounts of direct or indirect smoothness for curvature estimation, substantially controls the overall corner detection performance. We find three smoothing parameters in the literature: Gaussian smoothing-scale σ [16], chord-length L [6] and RoS k points on both sides of a curve-point [23]. Here, we call them smoothing-scale. While the first one is called a direct smoothing, the latter two are called indirect smoothing. A corner detector may use one [17] or more [6] of these three smoothing parameters.

In direct smoothing using σ [17], each curve-point P is convolved using a Gaussian function whose window size is determined by σ . While a large σ ensures a high degree of smoothness by involving a high number of neighboring points on both sides of P in the convolution operation, a small σ offers a low degree of smoothness. In indirect smoothing using L [6], the parameter value indicates how many neighboring points on both sides of P are considered during curvature estimation. In indirect smoothing using k [23], the parameter value indicates how many neighboring points on both sides of P are considered during the search for local curvature maxima.

In practice, real world images may have different types of corners. They contain important details belonging to a range of scales. Moreover, the *human visual system* (HVS) shows two characteristics [11]. Firstly, human visual perception treats images on several scales simultaneously. Secondly, when the signal to noise ratio decreases in an image, the HVS automatically increases the scales at which it characterizes the image. Therefore, it is preferable to detect corners using a multi-scale analysis.

Consequently, there are different kinds of detectors depending on the selection of their smoothing-scales:

- *Single-scale selection*: Detectors using a single smoothing-scale [1] suffer from the problem of selecting an appropriate value for σ , L or k for a given curve. The reason is that there may be different types and sizes of corners on a single curve. For examples, if σ is small, many spurious points are detected as corners. On the other hand, if σ is large, important details may be smoothed out (since curve becomes flat and curvature values decrease). In both of the cases, the detected corners may also be ill-positioned [30].
- *Full range of multi-scale selection*: In order to detect different types of corners, multi-scale corner detectors

have been proposed [17]. Detectors using a full range of scales [20] are computationally too expensive. The need to combine corners detected in different scales is a further problem.

- *Small range of multi-scale selection:* In order to avoid the heavy computational load associated with use of a full range of smoothing-scales, recently developed contour-based multi-scale detectors [17] detect corners on each curve using a fixed or one of three medium smoothing-scales. They then apply thresholds to remove weak and spurious corners. Finally, they may track the detected corners to improve their localization [17, 15].
- *Significant-scales selection:* Rosin [24] represented curves at their natural (significant) scales to avoid the expensive representation of each curve over a full range of scales. This representation describes each part of a curve at its significant scale (for which certain criteria are met). As a result, each curve is represented using many significant scales, instead of using the full range of scales.

Gao et al. [10] proposed a multi-scale corner detector based on local natural scale. However, local approaches are computationally more expensive and coarse scale features cannot be effectively detected using the local curve nature [24]. Beau and Singer [8] suggested only coarse feature detection in the reduced resolution curve to reduce computational complexity, because large scale features are more stable in image transformations and can be better used in different applications.

- *Adaptive-scales selection:* There are also adaptive smoothing techniques (also known as non-linear filtering or anisotropic diffusion of the signal) [21] for corner detection. The idea behind adaptive smoothing is to apply a versatile operator which can adapt to the local topography of the signal. This is achieved by repeatedly convolving the signal with a very small averaging mask weighted by a measure of the signal continuity at each point. The weights which are computed as a function of the pixel gradient vary at each pixel position and at each iteration.

Although the adaptive technique improves the localization of the detected corners without following a tracking step, such an iterative adaptation of the smoothing-scale is computationally demanding.

3.2. Curvature Estimation

Curve-point locations are quantized and may be affected by noise introduced during an image processing operation,

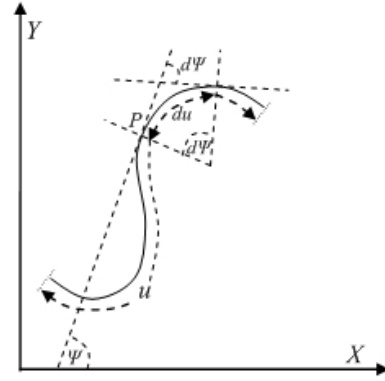


Figure 2. Intrinsic definition of curvature.

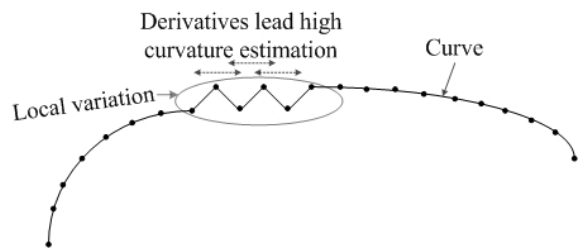


Figure 3. Derivative-based (direct) curvature estimation within a local variation of a curve.

as well as by the edge extractor. In addition, a curve may contain different sized corners which may require different amounts of smoothing. However, choosing or calculating an appropriate value for σ , L or k is very difficult for a given curve. As a result, no strict mathematical definition of curvature exists for a discrete curve [30] and curvature is only approximated [30].

As discussed in Section 2, there are two main types of curvature estimation techniques – direct and indirect [30]. In this section, we discuss them in detail.

3.2.1 Direct curvature estimation

Early dominant point detectors estimated angle as a significant measure (curvature indicated by cosine angle or tangential deflection) at each point P of the curve by considering a fixed RoS k . However, many modern dominant point detectors calculate a variable k on both sides of P with the expense of an additional computational cost. Depending on the above RoS determination criterion, dominant point detection algorithms can be divided into two groups. Firstly, algorithms with non-adaptive RoS keep k fixed for all curves and points [22, 23]. Secondly, algorithms with adaptive RoS calculate RoS for each point using the local nature of the curve [25].

Many corner detectors [20, 17] use Euclidean curvature. The Euclidean curvature at a point P is defined as the in-

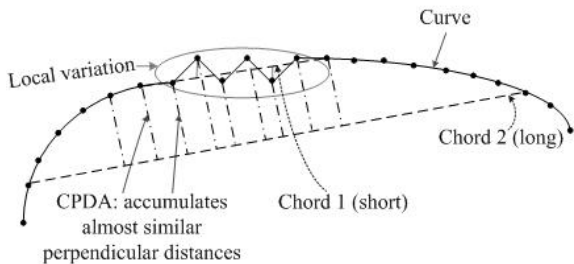


Figure 4. CPDA (indirect) curvature estimation within a local variation of a curve.

stantaneous rate of change of ψ , that is the angle subtended by the tangent at P with the x -axis, with respect to the arc-length u [20] (see Fig. 2). This curvature estimation technique considers a very small neighborhood (2×2) on both sides of P to evaluate the curvature at P [6].

Consequently, the estimated curvature using the above definition is very sensitive to the local variation and the noise on the curve. In a region with a high local variation, ψ changes significantly from point to point within a short curve segment. As depicted in Fig. 3, in a small but highly variable curve-region, the derivatives of the curve point-locations may lead to a high curvature estimation. As a result, detectors using the direct curvature estimation may detect many weak and false corners, if such local variation and noise is not smoothed away using a high smoothing-scale beforehand. However, smoothing has its own problems as we discussed in Section 3.1.

Detectors using cosine angle or tangential deflection [22] were found more sensitive to noise and less robust than those [17] using the Euclidean curvature [6].

3.2.2 Indirect curvature estimation

Since direct curvature estimation techniques are sensitive to noise and local variation of the curve, there are a number of detectors which use different curvature estimation techniques. For example, the detector in [9] used the distance from a curve-point to a chord of length L as a significance measure on that point. The CPDA detector [6] followed a distance accumulation technique using three chords as shown in Fig. 4. Since it uses a large neighborhood, it is less sensitive to noise and local variations on the curve [6]. Particularly, chord 1 (short) accumulates (sums) both positive and negative distances and chord 2 (long) accumulates distances which are almost the same for all points inside and outside this region. So the curvature product from the accumulated distances using three chords of different lengths is almost the same for all the points in and out of the local variation region. As a result, the CPDA detector detects corners with a higher average repeatability.

3.3. Integration of Smoothing and Curvature Estimation

Detectors using direct curvature estimation techniques may use no smoothing [23], direct smoothing only [17, 28, 9] or both direct and indirect smoothing [13]. Similarly, detectors using indirect curvature estimation techniques may use direct smoothing only [28], indirect smoothing only [9] or both [6].

Detectors estimating indirect curvature usually integrate indirect smoothing technique with curvature estimation [9, 6]. Zhang et al. [28] estimated indirect curvature without integrating or using an indirect smoothing technique.

Direct smoothing is mainly done before curvature calculation [17, 28, 6, 16] and indirect smoothing is usually used during curvature estimation. The aim of both of these smoothing techniques is to reduce the effect of noise and local variations [6]. In addition to a direct smoothing before curvature estimation, many detectors [15, 13, 23] also used direct [15] or indirect smoothing [13, 23] after curvature calculation. Mokhtarian and Mohanna [15] applied direct smoothing on the estimated curvature function and He & Yung [13] and Kitchen and Rosenfeld [23] applied an indirect smoothing (RoS k) for obtaining local curvature maxima points.

4. Performance of Promising Corner Detectors

Many of the contour-based detectors [17, 20] did not use any evaluation metrics. Instead they showed the detected corners on a small number of artificial and real images. In this section, we first summarize the experimental results from some recent papers [16, 27, 13, 5, 6, 28, 7]. Since different authors evaluated them using different evaluation metrics on different sets of databases, we then present an empirical study on a number of popular corner detectors using the automatic performance evaluation system [5] on a large database.

4.1. Summary of Existing Evaluations

Table 1 compares the evaluation systems and test results for contour-based detectors in the literature. We see that while many of the existing evaluation systems are manual (M), that means they involved human judgement, Awrangjeb and Lu [5, 6] introduced the automatic evaluation system (A) and included geometric sheared and lossy JPEG compressed images into the test dataset. Many authors [16, 27, 13] used a small set of images and He & Yung [13] did not use any geometrically transformed or signal processed images. This makes their evaluation outcomes less reliable. While compared with five most promising detectors, Awrangjeb and Lu showed that their CPDA detector [6] offered the best effectiveness in terms of both *average repeatability* and *localization error*.

Table 1. Comparisons of evaluation systems and test results for contour-based detectors in the literature. [Type: M = manual evaluation system involving human judgement [16], A = automatic evaluation system [6]; Error: maximum distance (in pixels) between reference and detected corners for a repeated corner, NA = not applicable; Database: number of original and test images; Attacks: how test images were generated, US = uniform scaling, NUS = non-uniform scaling; Test results: CCN = Consistency of corner numbers [16], Acu = Accuracy [16], TDR = true detection rate [27], FDR = false detection rate [27], L_e = localization error [6], EI = error index [28], R_{avg} = average repeatability [6].]

Year [Ref.]	Type	Error	Database		Attacks		Detectors	Test results		
			Ori.	Test	Geometric	Signal		CCN	Acu	
2006 [16]	M	3	5	590	Rotation(90)		CSS [17] ECSS [15]	CCN	Acu	
					US(50)			51	72	
					NUS(150) Affine(300)			60	78	
2007 [27]	M	3	5	1365	Rotation(90)		CSS [17] He & Yung [13] MSCP [27]	CCN	Acu	
					US(75)			47	75	
					NUS(400)			59	77	
					Affine(800)			73	84	
2008 [13]	M	4	2	-		CSS [17] ECSS [15] He & Yung [13]	TDR	FDR	L_e	
							93	21	1.53	
							92	14	1.33	
2008 [5]	A	3	23	8694	Rotation(414)	JPEG(460)	CSS [17] ECSS [15] ARCSS [5]	R_{avg}	L_e	
					US(345)	Noise(230)		64	1.36	
					NUS(2691)			58	1.38	
					Affine(3450) Shear(1104)			66	1.30	
2008 [6]	A	3	23	8694	Rotation(414)	JPEG(460)	RJ [22] CSS [17] ARCSS [5] He & Yung [13] MSCP [27] CPDA [6]	R_{avg}	L_e	
					US(345)	Noise(230)		52	1.37	
					NUS(2691)			65	1.36	
					Affine(3450)			66	1.30	
					Shear(1104)			70	1.23	
								71	1.26	
2009 [28]	M	NA	20	7320	Rotation(320)	Noise(200)	CSS [17] Zhang [28]	Acu	EI	
					US(200)			75	40	
					NUS(2200) Affine(4400)			78	33	
2009 [7]	A	3	23	8694	Rotation(414)	JPEG(460)	CSS [17] ARCSS [5] He & Yung [13] MSCP [27] CPDA [6] Fast-CPDA [7]	R_{avg}	L_e	
					US(345)	Noise(230)		64	1.36	
					NUS(2691)			65	1.30	
					Affine(3450)			69	1.23	
					Shear(1104)			70	1.25	
								75	1.14	
								74	1.15	

Since evaluation results in Table 1 were carried out on different platforms (different evaluation systems using different datasets and metrics), in the next section we have carried out an automatic evaluation of seven popular corner detectors using – *average repeatability* and *localization error* for effectiveness evaluation [6] and *running time* for efficiency evaluation. We aim to find out robust and efficient corner detectors.

4.2. Performance Comparison

In our performance study, we follow the automatic evaluation system introduced in [5], rather than the manual system in [16], which involves human judgement of corner detection and thereby is not suitable for the performance evaluation when using a large dataset.

In the automatic system, corners detected in the original images by a detector are considered as reference corners

and corners detected by the same detector in the test images which were signal processed and geometrically transformed, are taken as test corners. Then, reference and test corners are compared to evaluate the performance of that detector.

We evaluated seven detectors namely CSS [17], ARCSS [5], He & Yung [13], MSCP [27], Zhang et al. [28], CPDA [6] and Fast-CPDA [7] detectors. All the detectors were set at the default parameter settings indicated by their authors.

The database had a total of 25 different original 512×512 grey-scale images including some artificial images like ‘Block’ and real world images like ‘Lena’, ‘Leaf’, ‘House’ and ‘Lab’. All of the above original images were collected from standard databases [18, 26, 3] and corner detection examples are available at [3]. The database had a total of 9450 transformed (test) images, which were obtained by applying the following seven approaches of attacks on each original image:

- ◇ rotation at 18 different angles θ in $[-90^\circ, +90^\circ]$ at 10° apart, excluding 0° ;
- ◇ uniform (U) scaling factors $s_x = s_y$ in $[0.5, 2.0]$ at 0.1 apart, excluding 1.0;
- ◇ non-uniform (NU) scaling factors s_x in $[0.7, 1.3]$ and s_y in $[0.5, 1.8]$, at 0.1 apart, excluding the cases when $s_x = s_y$;
- ◇ combined transformations (rot.-scale): θ in $[-30^\circ, +30^\circ]$ at 10° apart, excluding 0° , followed by uniform or non-uniform scaling factors s_x and s_y in $[0.8, 1.2]$ at 0.1 apart;
- ◇ JPEG lossy compression at 20 quality factors in $[5, 100]$, at 5 apart;
- ◇ zero mean white Gaussian noise at 10 variances in $[0.005, 0.05]$ at 0.005 apart; and
- ◇ shearing factors s_{hx} and s_{hy} in $[0, 0.012]$ at 0.002 apart, excluding the one when $s_{hx} = s_{hy} = 0.0$.

Therefore, the database had a total of 450 rotated, 375 uniform scaled, 2925 non-uniform scaled, 3750 rotated and scaled transformed images. It also had 500 JPEG compressed, 250 Gaussian noise induced and 1200 sheared images. Note that transformations comprising rotations were also followed by cropping that removed the outer black parts. Consequently, many detected corners in the original images were cropped off in the test images for the transformations involving rotations.

From the evaluation results presented below we see that while the detectors using indirect curvature estimation techniques [6, 28] are more robust, the detectors using direct curvature estimation techniques [27, 17] are more efficient.

Table 2. Ranking of detectors using *average repeatability* (R_{avg} in percentage) and *localization error* (L_e in pixels).

Ranks	Detectors (R_{avg})	Detectors (L_e)
1	CPDA [6] (74.74)	CPDA [6] (1.14)
2	Zhang [28] (74.14)	Fast-CPDA [7] (1.15)
3	Fast-CPDA [7] (74.10)	Zhang [28] (1.20)
4	MSCP [27] (70.49)	He & Yung [13] (1.23)
5	He & Yung [13] (69.97)	MSCP [27] (1.26)
6	ARCSS [5] (65.36)	ARCSS [5] (1.30)
7	CSS [17] (64.44)	CSS [17] (1.36)

Table 3. Ranking of detectors using *running time* (per image).

Ranks	Detectors (sec.)
1	Fast-CPDA [7] (0.0048)
2	MSCP [27] (0.0056)
3	CSS [17] (0.0084)
4	He & Yung [13] (0.0090)
5	CPDA [5] (0.0131)
6	Zhang [28] (0.0146)
7	ARCSS [5] (0.0156)

4.2.1 Effectiveness

As shown in Table 2, the CPDA detector performed the best among the seven detectors in terms of both *average repeatability* rate and *localization error*. The Fast-CPDA [7] and Zhang et al. [28] detectors performed close to the CPDA detector.

From Table 2, it is evident that the detectors using direct curvature estimation techniques (CSS [17], ARCSS [5], He & Yung [13] and MSCP [27]) are less robust (effective) than those using indirect curvature estimation techniques (CPDA [6], Fast-CPDA [7] and Zhang et al. [28]).

4.2.2 Efficiency

Table 3 compares the running time of different detectors on a Windows XP machine with 3.00GHz of Intel(R) Core(TM)2 Duo CPU and 3.23GB of RAM.

We see that the Fast-CPDA detector [7] is the fastest among the seven followed by the MSCP detector [27]. The reason is that the Fast-CPDA detector estimates curvature only on a chosen limited number of curve-points and the MSCP detector simply applies a curvature-threshold to the curvature product function and both of them do not follow any corner tracking step. On the other hand, He & Yung [13], though a single-scale detector, is slow because it spent much time in adaptive curvature-threshold calculation using dynamic RoS. The ARCSS detector [5] is the slowest detector, since unlike the others it required significant time for affine-length parameterization of the curve. It also tracked the detected corners.

5. Conclusion

In this paper, we have carried out a comparative study of promising contour-based detectors. Experimentally, we have observed that while the detectors using indirect curvature estimation techniques are more robust, the detectors using direct curvature estimation techniques are faster.

The CSS-based detectors [16, 27, 13] suffer from two problems: inappropriate smoothing-scale (σ) selection and use of derivative-based curvature estimation [6]. The CPDA [6], Fast-CPDA [7] and Zhang et al. [28] detectors overcome these problems to a great extent. However, they are more robust but less efficient than their CSS-based counterparts. The future work includes investigating more efficient and effective indirect curvature estimating multi-scale detectors.

References

- [1] N. Ansari and K. W. Huang. Non-parametric dominant point detection. *Pattern Recognition*, 24(9):849–862, Sep 1991.
- [2] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):2–14, Jan 1986.
- [3] M. Awrangjeb. Image database and corner detection: <http://personal.gscit.monash.edu.au/awran/images.html>.
- [4] M. Awrangjeb. Contour-based corner detection and robust geometric point matching techniques. PhD Thesis, Monash University, Oct. 2008.
- [5] M. Awrangjeb and G. Lu. An improved curvature scale-space corner detector and a robust corner matching technique for transformed image identification. *IEEE Trans. Image Process.*, 17(12):2425–2441, Dec 2008.
- [6] M. Awrangjeb and G. Lu. Robust image corner detection based on the chord-to-point distance accumulation technique. *IEEE Trans. Multimedia*, 10(6):1059–1072, Oct 2008.
- [7] M. Awrangjeb, G. Lu, C. S. Fraser, and M. Ravanbakhsh. A fast corner detector based on the chord-to-point distance accumulation technique. In *Proc. Digital Image Computing: Techniques and Applications*, pages 519–525, Melbourne, Australia, Dec 2009.
- [8] V. Beau and M. Singer. Reduced resolution and scale space for dominant feature detection in contours. *Pattern Recognition*, 34(2):287–297, Feb 2001.
- [9] M. A. Fischler and R. C. Bolles. Perceptual organization and curve partitioning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):100–105, Jan 1986.
- [10] X. Gao, F. Sattar, A. Quddus, and R. Venkateswarlu. Multi-scale contour corner detection based on local natural scale and wavelet transform. *Image and Vision Computing*, 25(6):890–898, Jun 2007.
- [11] X. Gao, W. Zhang, F. Sattar, R. Venkateswarlu, and E. Sung. Scale-space based corner detection of gray level images using plessey operator. In *Proc. International Conference on Information, Communications and Signal Processing*, Bangkok, Thailand, Dec. 2005.
- [12] D. S. Guru, R. Dinesh, and P. Nagabhushan. Boundary based corner detection and localization using new ‘cornerity’ index: a robust approach. In *Proc. Canadian Conference on Computer and Robot Vision*, Ontario, Canada, May 2004.
- [13] X. C. He and N. H. C. Yung. Corner detector based on global and local curvature properties. *Optical Engineering*, 47(5):057008–1–057008–12, May 2008.
- [14] A. Masood and M. Sarfraz. Corner detection by sliding rectangles along planar curves. *Computers & Graphics*, 31(3):440–448, Jun 2007.
- [15] F. Mokhtarian and F. Mohanna. Enhancing the curvature scale space corner detector. In *Proc. Scandinavian Conference on Image Analysis*, pages 145–152, 2001.
- [16] F. Mokhtarian and F. Mohanna. Performance evaluation of corner detectors using consistency and accuracy measures. *Computer Vision and Image Understanding*, 102(1):81–94, Apr. 2006.
- [17] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(12):1376–1381, Dec. 1998.
- [18] F. A. P. Petitcolas. Photo database, http://www.petitcolas.net/fabien/watermarking/image_database/index.html.
- [19] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, Nov 1972.
- [20] A. Rattarangsi and R. T. Chin. Scale-based detection of corners of planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(4):430–449, Apr. 1992.
- [21] B. K. Ray and R. Pandyan. ACORD—an adaptive corner detector for planar curves. *Pattern Recognition*, 36(2003):703–708, 2003.
- [22] A. Rosenfeld and E. Johnston. Angle detection on digital planar curves. *IEEE Trans. Comput.*, C-22:875–878, Sep 1973.
- [23] A. Rosenfeld and J. Weszka. An improved method of angle detection on digital curves. *IEEE Trans. Comput.*, C-24:940–941, Sep 1975.
- [24] P. L. Rosin. Determining local natural scales of curves. *Pattern Recognition Letters*, 19(1):63–75, May 1998.
- [25] C. H. Teh and R. T. Chin. On the detection of dominant points on digital curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(8):859–872, Aug 1989.
- [26] USC-SIPI. The usc-sipi image database, <http://sipi.usc.edu/database/>.
- [27] X. Zhang, M. Lei, D. Yang, Y. Wang, and L. Ma. Multi-scale curvature product for robust image corner detection in curvature scale space. *Pattern Recognition Letters*, 28(2007):545–554, 2007.
- [28] X. Zhang, H. Wang, M. Hong, L. Xu, D. Yang, and B. C. Lovell. Robust image corner detection based on scale evolution difference of planar curves. *Pattern Recognition Letters*, 30(1):449–455, Mar 2009.
- [29] X. Zhang and D. Zhao. A parallel algorithm for detecting dominant points on multiple digital curves. *Pattern Recognition*, 30(2):239–244, Feb 1997.
- [30] P. Zhu and P. M. Chirlian. On critical point detection of digital shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):737–748, Aug 1995.